Collaborative CBR-based agents in the preparation of varied training lessons

Julien Henriet

Collaborative CBR-based agents in the preparation of varied training lessons

Julien Henriet

Université de Franche-Comté, Chrono-Environnement UMR 6249 CNRS, FEMTO-ST DISC UMR 6174 CNRS, 16 Route de Gray, 25000 Besançon, France

Abstract

Case-Based Reasoning (CBR) is widely used as a means of intelligent tutoring and e-learning systems. Indeed, course lessons are elaborated by analogy: this kind of system produces sets of exercises with respect to student level and class objective. Nevertheless, CBR systems always result in the same solution to a given problem description, whereas teaching requires that monotony be broken in order to maintain student motivation and attention. This is particularly true for sports where trainers must propose different exercises to practice the same skills for many weeks. We designed a system based on CBR that takes into account any previous lessons offered and designs new ones so as to vary the exercises each time: this system takes into account the solutions previously proposed so as to avoid giving the same lesson twice. In addition, this system is based on collaborative agents, each taking into account the exercises proposed by others so that each activity is proposed only once during a lesson. A sports trainer tested and evaluated the ability of this system as a means to design varied aïkido training lessons and proved that our system is capable of creating classroom activities that are diverse, changing, pertinent and consistent.

KEYWORDS: CASE-BASED REASONING, MULTI-AGENT SYSTEM, COLLABORATION, EDUCATION SYSTEM, SPORTS TRAINING.

Introduction

One of the major challenges in teaching is to maintain student motivation. Monotony and repetition of the same exercises contribute to boredom. On the contrary, originality and exercise diversity will challenge students and maintain their interest, even if the same aspect is practised during many class sessions. This is particularly true in sports where trainers have to propose varied exercises while having to train for the same skill during a block of weeks. Most of the tools provided by computer science, particularly from Artificial Intelligence (AI), would nevertheless give exactly the same exercises and lessons for training in a single given skill. In this particular domain, repetitive activities are a drawback, yet lesson planning is a process based on adaptation of past experiences. This paper presents a multiagent system that uses Case-Based Reasoning (CBR) systems which can provide lessons with widely differing progressions. CBR is a problem solving method that adapts the solutions from similar known problems in order to solve new problems (Kolodner J. , 1993). It stores cases called *source cases* and composed of problem and solution parts. Problems that occur and must be solved are called *target cases*. CBR describes a target case problem, retrieves the source cases whose problem parts most resemble those of the target case, reuses (adapts) the solutions of these most similar source cases, revises the adapted source case and capitalises this new experience in the system's knowledge base. CBR is widely employed in Intelligent Tutoring Systems (ITS) and e-learning systems (Graesser, Conley, & Olney, 2012). It is actually well-suited to this kind of system (Jamsandekar & Patil, 2013), as well as other tools from AI-like MultiAgent Systems (MAS) (Rishi, Govil, & Sinha, Agent based student modeling in distributed CBR based intelligent tutoring system, 2007), Artificial Neural Network (ANN) (Baylari & Montazer, 2009) and Genetic Algorithm (GA) (Tan, Shen, & Wang, 2012). J. Kolodner (Kolodner, Cox, & Gonzales-Calero, Case-based reasoning-inspired approaches to education, 2005) distinguished two types of CBR-inspired approaches to education: Goal-Based Scenarios (Schank, Fano, Bell, & Jona, 1994) where learners achieve missions in simulated worlds, and Learning By Design (Kolodner, Owensby, & Guzdial, Case-based learning aids, 2004) in which learners design and build working devices and obtain feedback thus confronting themselves with the real world. O.P. Rishi *et al.* designed an ITS based on agents and a CBR system (Rishi, Govil, & Sinha, Distributed case-based reasoning for intelligent tutoring system: An agent based student modeling paradigm, 2007) in which a *Personal Agent* is responsible for determining student level. A *Teaching Agent* then determines the education strategy with the help of a CBR regarding the description of the student level transmitted. Finally, a *Course Agent* provides and revises the lessons and exercises corresponding to the strategy proposed by the system with the help of a tutor. The CBR and GA based e-learning system proposed by Huang *et al.* also provides lessons taking into account the curriculum and the incorrect response patterns of a pre-test given to the learner (Huang, Huang, & Chen, 2007). A. Baylari and Gh. A. Montazer focused on the adaptation of tests to obtain a personalised estimation of a student's level (Baylari & Montazer, 2009). They used an ANN to correlate the student's answers to the tests and the exercises proposed by the teachers. Nevertheless, these approaches are insufficient for our system which seeks to change the proposed exercises. The systems in these approaches always correlate the same exercises to a single objective and learner level/experience. Our application domain requires that a variety of solutions be proposed for a given problem, taking into account solutions previously proposed. Indeed, when an athletic trainer wants to prepare an athlete, he/she must take care to maintain the latter's motivation. Thus, the proposed training sessions and exercises must vary, since the same exercises practiced time after time would be boring. Finally, our system is based on agents: a lesson has one objective which is divided into sub-objectives and each agent is in charge of one sub-objective. This introduces another difficulty since an exercise must not be proposed more than once in the same lesson. Thus, solutions proposed by all the agents must be chosen collectively, taking into account the training history (the previous lessons proposed to the athlete during the season) as well as the solutions proposed by all other agents. Our application is therefore a MAS based on collaborative agents. E. Plaza and L. McGinty presented different policies suited to different cases of distributed CBR systems (Plaza & Mc Ginty, 2005). Nevertheless, these policies are well-suited to determining which solution is the best, considering a set of concurrent proposed solutions. Each agent of our system must provide solutions drawn from a set of common exercises, solutions to different problems and without proposing exercises chosen by the other agents. In the next section, we present the architecture of the distributed system we have designed. Its implementation and performance are then presented and analysed.

Methods

In the first part of this section, we detail the lesson structure of our distributed system. The distributed architecture and the data flows are examined in the second part. Finally, in the third part, we present how a lesson is designed theoretically, with an example to illustrate.

Lesson structure

To coach an athlete for a competition, the trainer has to make him/her improve different aspects before this particular appointment. The trainer divides the course into cycles. A cycle usually takes three to seven weeks, and there are a minimum of three lessons per week. During each cycle, the trainer emphasises one particular aspect. Thus, over many lessons, the trainer sets the same objective (Matveev, 1965), (Issurin, 2010), (Garcia-Pallares, Garcia-Fernandez, Sanchez-Medina, & Izquierdo, 2010), (Ronnestad, et al., 2012). In sports, an objective is usually expressed in the following manner: "the athlete(s) must become capable of doing something.". This objective can be technical, tactical, physical or sometimes even psychological. Such an objective is then divided into different sub-objectives which are elementary and that permit the athlete to reach the main objective. Many lessons are usually necessary in order to work and reach all the sub-objectives of a single objective. In addition, one sub-objective can be shared by two or more objectives. Going a step further, the same relations and constraints exist between sub-objectives and exercises: to reach one sub-objective, the trainer will choose and prepare different exercises from among a set of known ones, and a given exercise can contribute to reaching many other sub-objectives. If we take the example of aïkido training, which is a Japanese martial art, one objective may be for the aïkidoka, when grabbed, to become capable of constructing an aïkido technique." Here are some of the numerous possible sub-objectives: the aïkidoka becomes capable of pivoting although grabbed, moving although grabbed, breaking the posture of a partner, relaxing the arms despite being strongly grabbed, using the grip to unbalance the partner, defining a meeting point with a partner, etc. Thus, aïkido is based on martial techniques that will emphasise a particular point and therefore allow one to understand, practice and reach each sub-objective listed above. An exercise consists of practising one technique following the trainer's instructions. A technique will require different skills, so teachers can propose the same technique for different sub-objectives. The trainer's instructions will then stress one particular aspect of the technique regarding the sub-objective to be reached. We have created two types of CBR, thus our system deals with four types of case:

- A CBR that proposes sub-objectives regarding objectives:
  - Source cases noted $s=(O, \cup\{(SO, D^{S}_{SO,O})\})$ where objective O is the problem part of s, and the set of sub-objectives SO with the duration SO must be practised during the season (noted $D^{S}_{SO,O}$) in order to reach O, and is the solution part of s noted $\cup\{(SO, D^{S}_{SO,O})\}$;
  - Target cases noted $t=(O, \cup\{(SO, D^{T}_{SO,O})\})$;
- CBRs that propose exercises regarding sub-objectives:
  - Source cases noted $\sigma=(SO, \cup\{(EX, INSTR_{EX,SO}, CD^{\sigma}_{EX,SO}, RD^{\sigma}_{EX,SO})\})$ where SO is the problem part of σ, and the duration EX must be practised when proposed (noted $CD^{\sigma}_{EX,SO}$) and the duration EX must be practised during the season in order to reach SO (noted $RD^{\sigma}_{EX,SO}$), is the solution part of σ;
  - Target cases noted $\tau=(SO, \cup\{(EX, INSTR_{EX,SO}, CD^{\tau}_{EX,SO}, RD^{\tau}_{EX,SO})\})$.

Actually, durations $D^{S}_{SO,O}$ and $RD^{\sigma}_{EX,SO}$ give the priorities of each sub-objective and exercise, priorities which increase with the time available. The initial durations are given by the trainer, but further study will base these initial values on pre-tests as proposed in different approaches (Tan, Shen, & Wang, 2012), (Huang, Huang, & Chen, 2007). The system must be initialised each year, at the beginning of the season. Further investigation will focus on its initialisation process.

System architecture and communication model

MAS constitute a paradigm designed to handle distributed systems. They are the product of AI research and reflect its limits: a single AI representing the behaviour of a unique entity cannot deal with collective behaviour. The idea of distributing the intelligence thus appears and one can speak of Distributed Artificial Intelligence (DAI). In a MAS, an agent is a physical or abstract entity with some specific characteristics: a perception of its environment (including itself and the other agents), a capability to act (upon itself or upon the environment) and an autonomy in its decisions and actions. To design a MAS is not only to design the different agents but the environment too. As explained in the previous section, the choice of the sub-objectives regarding an objective is an autonomous process, as well as the determination of the exercises regarding a sub-objective, the other exercises chosen and their priority level. Each process is based on specific rules and reasoning. In addition, it must interact with the other processes and take their choices into account. Thus, each process must be autonomous, make decisions, infer changes on the choices made by the others, be aware of its environment, communicate and interact with the others. Consequently, we can call them agents.

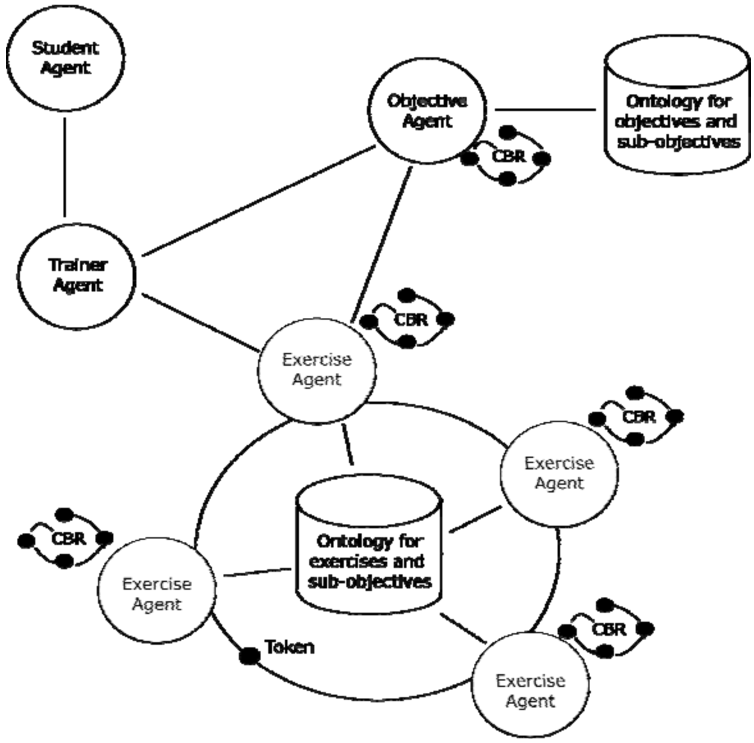

Figure 1. Platform communication model

As shown in Figure 1, the system is composed of four types of agent: the trainer agent, the student agents, the Objective Agent (OA) which is responsible for choosing the sub-objectives regarding an objective requested by the trainer, and the exercise agents which are each responsible for proposing the exercises the best suited to one sub-objective. One sub-objective is given by the OA or by another Exercise Agent (EA) to each EA. Each EA must also take into account the choices made by the other EAs: each exercise can be done only once during the entire lesson. Thus, the choices of the other EAs are shared and a decision policy is designed and tested. To share the memory among the EAs, we have chosen the Pilgrim protocol which is an efficient and secured protocol for concurrent, cooperative and collaborative works with shared memory (Garcia, Guyennet, Henriet, & Lapayre, Research approach advance in concurrency management algorithms for cooperative work, 2005). The agents are dispatched over a ring and a token is exchanged. The originality in this protocol is the fact that the token, carrying each stored modification, is transmitted to all of the other agents, thus each agent has a copy of the shared memory (i.e. the set of exercises proposed by the other agents). The decision policy is implemented over each EA that is able to modify its set of proposed exercises, if one or more of its exercises are identical to any of those found in another set.

Determination of sub-objectives

Once the lesson objective is chosen by the trainer, after having analysed any additional student needs, the objective agent is responsible for choosing the set of sub-objectives and their duration in function of those which have been done before and also the students' degree of assimilation. This choice is made by the OA according to the CBR approach.

Case model

For the OA, a case is a set, composed of two parts, a problem and a solution. As presented before, each problem part is composed of an objective O, and the solution part of a set of (SO, $D_{SO,O}$) where SO is a sub-objective. The durations $D_{SO,O}$ are updated during the capitalisation process taking into account the remarks made by the trainer during the revision phase. Since the students' levels of expertise increase, we can consider that durations decrease and thus we can speak of "remaining durations".

Retrieval phase

The retrieval phase is responsible for retrieving and sorting all the possible sub-objectives linked to a single objective. Thus, the retrieval phase consists of retrieving the source case for which the chosen objective is the problem part.

Adaptation phase

The adaptation phase consists of computing the durations of each sub-objective. Firstly, the adaptation module sorts the set of sub-objectives with their durations so that the sub-objective with the maximum duration comes first and the one with the minimum comes last. Then, the proposed durations of t are calculated. For this computation, the trainer has to give the desired number of sub-objectives, the total number of sub-objectives associated with one objective and the total duration of the course respectively noted N_{SO}, $Card(Sol_S)$ and D. The adaptation consists of giving the same duration to each desired sub-objective.

Thus, $t = (O, \cup_{n \in \{1..Card(Sols)\}}\{(SO_n, D^{T,n}_{SOn,O})\})$ where $D^{T,n}_{SOn,O} = \min(D^{S,n}_{SOn,O}, (D/N_{SO}))$ $\forall$ n $\in$ $\{1..N_{SO}\}$. And then filling the possible remaining time with sub-objectives of lower priority:

$\forall n \in \{(N_{SO}+1)..Card(Sol_S)\}$, if $(\Sigma_{i=1}^{n-1} D^{T,n}_{SOn,O}) < D)$

then $D^{T,n}_{SOn,O}$=min($D^{S,n}_{SOn,O}$, (D-$\Sigma_{i=1}^{n-1}D^{T,n}_{SOn,O}$)),

otherwise (SO_n , $D^{T,n}_{SOn,O}$) is removed from the solution part of *t*.

Revision phase

The revision phase begins just after the adaptation, before the lesson, and finishes after the lesson with remarks and evaluation of the students' acquired level of capacity. Consequently, the revision phase is one of the most important in our system since before the lesson, the trainer modifies (adds, updates, removes) the sub-objectives and durations, and after the lesson, the trainer evaluates the skills mastered by the students. After having modified the solution part of *t* and before the lesson begins, each element of *t* is transmitted to one EA that will have to associate the corresponding exercises. After the lesson, each sub-objective duration is modified: for each element of the solution part of *t*, $D^{T,n}_{SOn,O}$=$D^{T,n}_{SOn,O}$x($level_{SOn}$/10) where $level_{SOn}$ is the evaluation from 0 to 10 of the students level for the sub-objective SO_n. Thus, the remaining time associated with each sub-objective will decrease slowly as long as students do not reach the required level. In contrast, this remaining duration will decrease quickly when they reach the expected level rapidly.

Capitalisation phase

The first task consists of retrieving the source case *s* for which the selected objective *O* is the problem part, and adding the sub-objectives that do not yet exist (the ones that may have been added during the revision phase). The system then subtracts the durations of *t* from all the durations of the source cases: for all *SO* in the solution part of *t*, for all *s* for which *SO* appears in the solution part (even if its objective *Os* is not the selected objective *O*), $D^{S}_{SO,Os}$=min(0 , ($D^{S}_{SO,Os}$-$D^{T}_{SO,O}$)).

Example of how lesson sub-objectives are selected

This part presents the method through an example of how an aïkido training lesson is planned. Table 1 presents two source cases.

Source case	Objective	Sub-objective	Duration (minutes)
1	Using a grip	Breaking the partner's posture	100
		Relaxing despite a grip	100
		Making the partner lose balance	90
		Pivoting around a grip	90
2	Breaking a grip	Breaking a single grip	100
		Relaxing despite a grip	80

Table 1. Two source cases stored in the system.

Assuming, the trainer chooses the objective *O* ='Using a grip' as the main point of a lesson *D* = 90 minutes long and with a minimum of N_{SO} = 3 different sub-objectives. The trainer transmits these parameters to the OA. Obviously, the OA remains case 1. The adaptation process then sorts the sub-objectives according to their durations and designates the solution presented in Table 2, allocating *D* = 90/3 = 30 to each selected sub-objective duration. Hence, assuming the trainer has not modified the list of sub-objectives, the examples of the trainer's evaluations made during the revision process just after the course, are reported in the last column of Table 2. Consequently, after capitalisation, the new durations will be as reported in Table 3. Even if a sub-objective is associated with another objective, its duration is diminished for both. This is the case for the last associated sub-objective of source case 2. As shown in this table, the less assimilated sub-objectives ("Relaxing despite a grip" and "Pivoting around a grip") become the most prior ones. We also note that, as required for the system specification, if the same objective ("Using a grip") is selected again, the less assimilated sub-objective with other sub-objectives will be selected ("Relaxing despite a grip", "Pivoting around a grip", "Breaking the partner's posture"). Thus, as required, the proposed solutions will change even if the same objective is requested again later.

Sub-objective	Duration (minute)	Trainer's evaluation (/10)
Breaking the partner's posture	30	7
Relaxing despite a grip	30	3
Making the partner lose balance	30	4

Table 2. Chosen sub-objectives and their evaluation by the trainer.

Source case	Objective	Sub-objective	Duration (minutes)
1	Using a grip	Breaking the partner's posture	100-(30x7/10)=79
		Relaxing despite a grip	100-(30x3/10)=91
		Making the partner lose balance	90-(30x4/10)=78
		Pivoting around a grip	90
2	Breaking a grip	Breaking a single grip	100
		Relaxing despite a grip	80-(30x3/10)=71

Table 3. Cases stored in the ontology objective/sub-objective.

Selection of exercices

This sub-section presents how the exercices are chosen regarding the selected sub-objectives.

Case model

For this part of the system the problem part of a case is composed of a sub-objective while the solution part is composed of a set of exercises with a specified duration to spend on them. Though exercises have constant durations, for sub-objectives the total remaining duration required for each exercise must be specified and will decrease from one lesson to the next. Consequently, as presented previously, each source case $\sigma=(SO, \cup\{(EX, INSTR_{EX,SO}, CD^{\sigma}_{EX,SO}, RD^{\sigma}_{EX,SO})\})$ contains the exercises possible regarding *SO*. Assuming $Card(Sol_{\sigma})$ is the number of exercises of the solution part of σ, the target case τ_i taken into account by the EA EA_i is noted $\tau_i=(SO_i , \cup_{n\in\{1..Card(Sol\sigma)\}}\{(EX_n , INSTR_{EXn,SOi} , CD^{\tau i}_{EXn,SOi} , RD^{\tau i}_{EXn,SOi})\})$.

Retrieval phase

EA_1 is the EA initiated by the trainer. Hence, its role also consists of initiating as many EAs as required, and creating the token. Each EA_i then retrieves the source case corresponding to SO_i.

Adaptation phase

The adaptation phase requires the EAs to communicate with each other in order to associate exercises with sub-objectives. To accomplish this they use the Pilgrim protocol to share the sets (Garcia, Guyennet, Henriet, & Lapayre, Towards an optimistic management of concurrency: A probabilistic study of the Pilgrim protocol, 2006), the agents exchanging a token around a ring. The protocol's originality resides in the fact that the token transmits the modifications made to the shared memory. The number of exercises selected depends on the duration required: the exercises are decreasingly ordered according to their remaining duration and each one is added until the sum of constant durations reaches the required level for sub-objective SO_i, i.e. $(\Sigma_n CD^{\tau i}_{EXn,SOi}) \geq D^{T,i}_{SOi,O}$. Before placing its list of exercises on the token, EA_i verifies whether each exercise has been selected with a higher or equal duration by another EA. If so, EA_i removes its exercise and replaces it with the next unselected one. EA_1 acknowledges when all the sub-objectives have been provided with exercises and acknowledges again when the list is complete. Then, when EA_1 receives the token again, it verifies whether any modification has been made by the other EAs. The token continues to travel around the ring until EA_1 receives it with the complete list acknowledged and with no further modification. During the process of selection of SOs, the trainer may have created a new sub-objective which will not be associated with any exercise. If so, the EA can confirm that no exercise was found for this SO by using the exercise named *LackOfExercise* with the duration required. Also, the EA may not have enough remaining exercises to the duration period. Thus, it can use this *LackOfExercise* too.

Revision phase

The OA performs a revision phase, but the trainer can also create, modify, replace or remove exercises, their durations and their instructions before the lesson. After the course, the trainer evaluates student levels. The same computations as those applied to *SO* determination are applied to the exercise durations of the ontology:

$$\forall i \, , \forall n \in \{1..N^{EX}_i\}, CD^{\tau i}_{EXn,SOi} = CD^{\tau i}_{EXn,SOi} \times (level_{EXn}/10).$$

Capitalisation phase

As previously, during the capitalisation process, the possible new exercises are added to the corresponding cases. Then, the same kind of computation as before is applied to each exercise, whatever the associated *SO*: $\forall SO \, , \forall EX \, , RD^{\sigma}_{EX,SO}=\min(0 \, , (RD^{\sigma}_{EX,SO} - CD^{\tau}_{EX,SO}))$.

Example of selection of the exercices for one sub-objective

In this subsection we consider the example presented in the previous section and the selected sub-objectives with durations reported in Table 2. Table 4 presents some sets of aïkido techniques, called by their Japanese name, that correspond to each of the selected sub-objectives. For greater clarity we did not report the instructions for each exercise in the example, assuming they were of no help in understanding how the distributed system works. Thus, three EAs are called by the OA: EA_1 is the corresponding agent that has to treat the sub-objective 'Breaking the partner's posture'. It places the two remaining sub-objectives on the token which it transmits to EA_2, in charge of the sub-objective 'Relaxing despite a grip'. EA_2 then transmits the token with the last sub-objective to EA_3 which will have to manage 'Making the partner lose balance'. By the time the token returns to EA_1, the latter will have selected the techniques and durations reported in Table 5. The agent places them on the token and transmits it to EA_2. EA_2, upon receiving the token, has already selected first the techniques reported in Table 5. It is worth nothing that EA_1 and EA_2 have both chosen 'Katateryotedori Kokyuhoo', but since the remaining duration of this technique for the EA_2 sub-objective (50) is higher than the remaining duration for the EA_1 sub-objective (40), EA_2 places it and EA_1 is the one that will have to replace it. The token is then transmitted to EA_3. Before the token arrives, EA_3 has selected the techniques reported in Table 5. None of the selected techniques has been used by the other agents, thus EA_3 can place them on the token and transmit them to EA_1. When EA_1 receives the token, it remarks that it has to change one of the techniques that it proposed. Thus, it selects the most prior and available technique instead and adds a complete list acknowledgement to the token before transmitting it to EA_2. The techniques finally proposed by EA_1 are presented in Table 6. Since each technique figures only once, the lesson presented in Table 7 is transmitted to the trainer. After the lesson, the trainer must evaluate the assimilation of each technique proposed. Assuming the evaluations as reported in Table 8, the new remaining durations are computed for each exercise and stored in the ontology by EA_1.

Sub-objective	Aïkido technique	Constant duration (min.)	Remaining duration (min.)
Breaking the partner's posture	Ryotedori Tenchinage	10	50
Breaking the partner's posture	Katatedori Kokyuhoo	10	50
Breaking the partner's posture	Katateryotedori Kokyuhoo	10	40
Breaking the partner's posture	Aihamikatatedori Ikyo	10	40
Breaking the partner's posture	Aihamikatatedori Iriminage	10	30
Relaxing despite a grip	Katateryotedori Kokyunage	10	50
Relaxing despite a grip	Katateryotedori Kokyuhoo	10	50
Relaxing despite a grip	Katateryotedori Udekimenage	10	40
Making the partner lose balance	Katatedori Kokyunage	10	50
Making the partner lose balance	Aihaminkatatedori Ikyo	10	50
Making the partner lose balance	Ushiroryotedori Kokyunage	10	50

Table 4. Cases stored in the ontology sub-objective/exercice.

Agent	Aïkido technique	Constant duration (min.)	Remaining duration (min.)
EA_1	Ryotedori Tenchinage	10	50
EA_1	Katatedori Kokyuhoo	10	50
EA_1	Katateryotedori Kokyuhoo	10	40
EA_2	Katateryotedori Kokyunage	10	50
EA_2	Katateryotedori Kokyuhoo	10	50
EA_2	Katateryotedori Udekimenage	10	40
EA_3	Katatedori Kokyunage	10	50
EA_3	Aihamikatatedori Ikyo	10	50
EA_3	Ushiroryotedori Kokyunage	10	50

Table 5. Exercices initially selected by EA_1, EA_2 and EA_3.

Agent	Aïkido technique	Constant duration (min.)	Remaining duration (min.)
EA_1	Ryotedori Tenchinage	10	50
EA_1	Katatedori Kokyuhoo	10	50
EA_1	Aihamikatatedori Iriminage	10	30

Table 6. Modified list of exercices of EA_1.

Sub-objective	Aïkido technique	Duration (min.)
Breaking the partner's posture	Ryotedori Tenchinage	10
Breaking the partner's posture	Katatedori Kokyuhoo	10
Breaking the partner's posture	Aihamikatatedori Iriminage	10
Relaxing despite a grip	Katateryotedori Kokyunage	10
Relaxing despite a grip	Katateryotedori Kokyuhoo	10
Relaxing despite a grip	Katateryotedori Udekimenage	10
Making the partner lose balance	Katatedori Kokyunage	10
Making the partner lose balance	Aihamikatatedori Ikyo	10
Making the partner lose balance	Ushiroryotedori Kokyunage	10

Table 7. Exercices finally transmited to the trainer.

Sub-objective	Aïkido technique	Constant duration (min.)	Trainer evaluation (/10)	Remaining duration (min.)
Breaking...	Ryotedori Tenchinage	10	8	50 – 8 = 42
Breaking...	Katatedori Kokyuhoo	10	6	50 – 6 = 46
Breaking...	Katateryotedori Kokyuhoo	10		40 – 4 = 36
Breaking...	Aihaminkatatedori Ikyo	10		40 – 5 = 35
Breaking...	Aihamikatatedori Iriminage	10	7	30 – 7 = 23
Relaxing...	Katateryotedori Kokyunage	10	3	50 – 3 = 47
Relaxing...	Katateryotedori Kokyuhoo	10	4	50 – 4 = 46
Relaxing...	Katatryotedori Udekimenage	10	2	40 – 2 = 38
Making...	Katatedori Kokyunage	10	4	50 – 4 = 46
Making...	Aihamikatatedori Ikyo	10	5	50 – 5 = 45
Making...	Ushiroryotedori Kokyunage	10	3	50 – 3 = 47

Table 8. Cases stored in the ontology sub-objective/exercice after revision.

Results

This section presents the application's performances and how it was evaluated. The system was evaluated for both of the criteria for which it was designed, its ability to propose varied courses (sub-objectives and exercises), and to propose pertinent courses (sub-objectives and exercises). Hence, a qualified aïkido trainer evaluated 15 lessons proposed by the implemented system with the same objective ('Use a grip'). He previously entered the 6 corresponding sub-objectives (presented in Table 9) and their initial durations. He also entered 27 techniques aimed at the sub-objectives previously entered, along with their durations (cf Table 10). We listed the sub-objectives and techniques proposed by the system for each course. The trainer simulated the skill level of a group of students to whom the course was proposed. The distribution of the different sub-objectives among the courses is reported in Table 9.

All of the sub-objectives appear about the same number of times (7 and 8 times each). The sub-objectives most often chosen for the first three lessons ('Break partner's posture' and 'Do Irimi despite a grip') were the ones with the longest durations. This table clearly shows there is a variation and turn-over in the chosen sub-objectives. Nevertheless, during the first 6 lessons, the trainer gave the same score to all of the sub-objectives of each lesson. The use of different scores for the sub-objectives in the following lessons implied that these should be mixed and varied.

Lesson number	Break partner's posture	Do Irimi despite a grip	Move despite a grip	Pivot despite a grip	Relax despite a grip	Make lose balance
1	Chosen	Chosen	Chosen			
2	Chosen	Chosen		Chosen		
3	Chosen	Chosen	Chosen			
4				Chosen	Chosen	Chosen
5				Chosen	Chosen	Chosen
6	Chosen	Chosen	Chosen			
7				Chosen	Chosen	Chosen
8	Chosen	Chosen	Chosen			
9			Chosen		Chosen	Chosen
10		Chosen		Chosen		Chosen
11	Chosen		Chosen		Chosen	
12				Chosen	Chosen	Chosen
13	Chosen	Chosen	Chosen			
14				Chosen	Chosen	Chosen
15		Chosen	Chosen			Chosen

Table 9. Sub-objectives chosen by the system.

Aïkido technique	Number of times of appearance	Aïkido technique	Number of times of appearance
Aihamikatatedori Ikyo	7	Katadori Iriminage	3
Aihamikatatedori Iriminage	4	Katadori Kokyunage	7
Katatedori Ikyo	3	Katadori Shihonage	4
Katatedori Kokyuhoo	7	Maeryokatadori Iriminage	4
Katatedori Kokyunage	9	Maeryokatadori Kokyunage	1
Katadorimenushi Kotegaeshi	5	Ryotedori Ikyo	7
Katadorimenushi Kokyunage	5	Ryotedori Kotegaeshi	7
Katatedori Udekimenage	6	Ryotedori Tenchinage	7
Katateryotedori Ikyo	3	Ushirokatatedorikubishime Sankyo	3
Katateryotedori Kotegaeshi	2	Ushiroryotedori Ikyo	4
Katateryotedori Kokyuhoo	8	Ushiroryotedori Iriminage	5
Katateryotedori Kokyunage	7	Ushiroryotedori Kokyunage	6
Katateryotedori Shihonage	2	Yokomenushi Ikyo	4
Katateryotedori Udekimenage	5		

Table 10. Number of times each technique was chosen.

Table 10 shows the number of times each exercise (aïkido technique) was chosen. According to the aïkido teacher, some techniques are more fundamental than others to each sub-objective. The teacher gave a higher score to these techniques and they were actually chosen 6, 7, 8 or 9 times: 'Katatedori Kokyunage' to help students feel their way to making a partner lose his/her balance and how to pivot round a grip. 'Katateryotedori Kokyuhoo' is of great help in learning how to break a partner's posture and also how to move and relax despite a grip, etc. All techniques highly useful in learning multiple abilities are in this category, followed, to varying degrees, by those associated with just one sub-objective. As shown in this table, all of the techniques were selected once or more, which reveals a good turn-over of the exercises from one lesson to another. Techniques chosen once, twice or three times were those with the lowest scores and that appeared once in the sub-objectives with the lowest duration.

Figure 2 presents the scores given by the teacher to the proposed lessons. These scores are based on two criteria:

- The pertinence of the chosen sub-objectives regarding the lessons, along with previously given scores, and
- The pertinence of the exercises (techniques) determined by the system.

We also asked the trainer to take into account the variety of the sub-objectives and exercises. Ten points were given to very satisfying lessons, whereas poorly satisfying courses were given a score of 0. Every lesson was rated at more than 5 points by the trainer who gave a mean value of 7.6 to the entire course. As represented in this figure, the lowest marks were given to lessons 3, 5 and 8. They were qualified by the trainer as quite repetitive compared to the previous lessons generated by the system. Actually, this was due to the fact that during the revision processes the trainer had given the same score to all of the course's sub-objectives and exercises up to lesson 6. Then, with the introduction of differing scores for the sub-objectives, the lessons became more varied. Indeed, all lessons from 9 to 15 satisfied the trainer (8 or 9 points for these lessons).

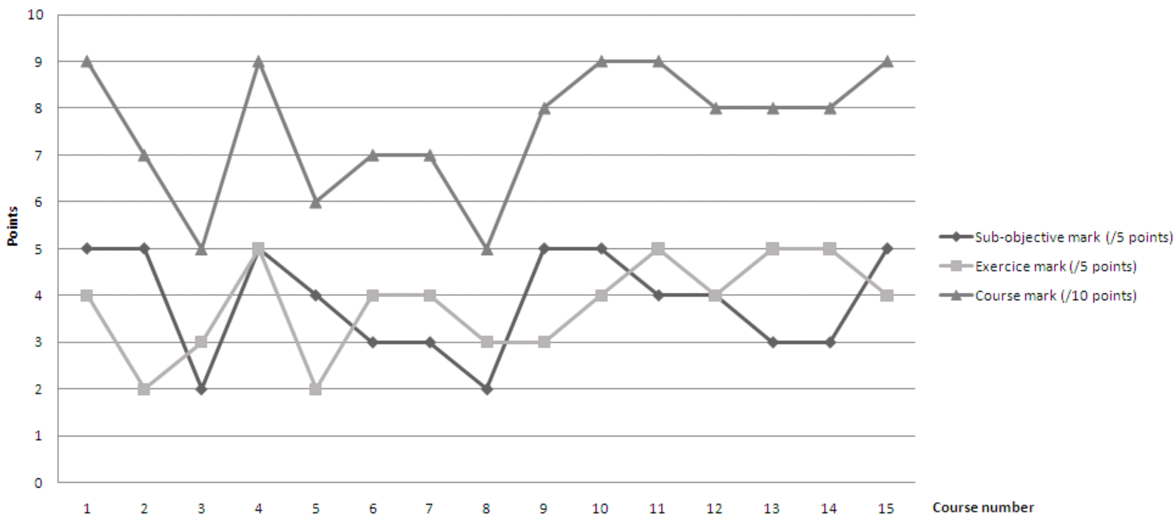

Figure 2. Lesson evaluations.

Discussion

The results presented above prove that by taking into account previous experiences, the system and our method are capable of proposing pertinent and varied lessons. An important part of our method resides in the agents' capacity in real time to take into account remarks made by users during the revision processes and also to adapt their own solutions to those concurrently provided by other agents of the distributed system. This was possible only due to the use of a concurrency management protocol designed to share real-time memory and concurrent actions. Indeed, broadcasting the solutions would have introduced concurrent and inconsistent data and lessons since there is no global schedule in collaborative and distributed systems. This particular aspect of distributed CBR systems will be examined through further study.

One limit to our system lies in the fact that it must be initialised each year at the beginning of the season or whenever there is no remaining duration left. At that time, the trainer must produce all of the initial values. Additional invetigation will propose a process based on experience in order to compute initial values. Most of the other approaches use pre-tests (Huang, Huang, & Chen, 2007), (Tan, Shen, & Wang, 2012) that evaluate students' levels. Another limit to our system is found in the way creations of sub-objectives are managed: In future study we will focus on the way an EA might propose exercises for a newly created sub-objective in function of the exercises proposed for similar sub-objectives. At the very least, the trainer must take time to again sort through the sub-objectives and to examine the order of the exercises proposed by the system: after having selected them, it may be of interest to take into account another parameter, closer to the nature of the exercises in order to better classify them.

Finally, like the approaches of A. Cordier *et al.* (Cordier, Fuchs, & Mille, 2006), S. Craw *et al.* (Craw, Wiratunga, & Rowe, 2006), J. Lieber *et al.* (Lieber, 2007), (Dufour-Lussier, Le Ber, Lieber, & Nauer, 2013) and the one we applied in EquiVox with adaptation using interpolation tools (Henriet, Leni, Laurent, & Salomon, 2014), ours establishes a link between adaptation and capitalization of revisions. Indeed, we have examined a way to use the remarks of users made during the revision phase in order to enhance the accuracy of the adaptation process of CBR-systems.

Conclusion

This paper presents a multi-agent system that can generate and personalise lessons. The design is based on agents that use CBR systems to propose varied courses. It emphasises the importance of the revision process and its consequence on the adaptation phase. During this revision process, the user furnishes information as to the level that students have attained while the system stores the proposed lesson in order to enhance the adaptation of future lessons. Though AI systems learn to always produce the same solutions to the same problems, our system is capable of proposing a variety of solutions. It is capable of proposing sports training lessons that are diverse, changing, pertinent and consistent, enabling students to practise the same given skill over many weeks.

Ours is a distributed system that shares memory. A collaboration protocol based on a token ring has been adapted and used, as well as a policy capable of merging and arbitrating between concurrent solutions proposed by the collaborative agents during their adaptation process.

Future study will focus on the enhancement of the initialisation process since the pertinence of the solutions proposed depends heavily on the initial values stored. Introducing processes such as the pre-testing of students may improve the quality of the adapted solutions generated.

References

Baylari, A., & Montazer, G. A. (2009). Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems With Applications , 36*, 8013-8021.

Cordier, A., Fuchs, B., & Mille, A. (2006). Engineering and learning of adaptation knowledge and Case-Based Reasoning. *Springer-Verlag* (pp. 303-317).

Craw, S., Wiratunga, N., & Rowe, R. C. (2006). Learning adaptation knowledge to improve case-based reasoning. *Artificial Intelligence , 170*, 1175-1192.

Dufour-Lussier, V., Le Ber, F., Lieber, J., & Nauer, E. (2013). Automatic case acquisition from texts for process-oriented case-based reasoning. *Information Systems , 40*, 153-167.

Garcia, E., Guyennet, H., Henriet, J., & Lapayre, J. C. (2005). Research approach advance in concurrency management algorithms for cooperative work. *International Journal of Computer Science and Network Security, Special issue Computer Science* , 1-12.

Garcia, E., Guyennet, H., Henriet, J., & Lapayre, J. C. (2006). Towards an optimistic management of concurrency: A probabilistic study of the Pilgrim protocol. (W. e. Shen, Éd.) *Computer Supported Cooperative Work in Design , 3865*, 51-60.

Garcia-Pallares, J., Garcia-Fernandez, M., Sanchez-Medina, L., & Izquierdo, M. (2010). Performance changes in world-class kayakers following two different training periodization models. *European Journal of Applied Physiology , 110*, 99-107.

Graesser, J. L., Conley, M. W., & Olney, A. (2012). Intelligent tutoring systems. (K. R. Harris, S. Graham, & T. Urdan, Éds.) *The APA Educational Psychology Handbook , 3*, 451-473.

Henriet, J., Leni, P. E., Laurent, R., & Salomon, M. (2014). Case-based reasoning adaptation of numerical representations of human organs by interpolation. *Expert Systems With Applications , 40*, 260-266.

Huang, M. U., Huang, H. S., & Chen, M. Y. (2007). Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems With Applications , 33*, 551-564.

Issurin, V. (2010). New horizons for the methodology and psychology of training periodization. *Sports medicine , 40* (3), 189-206.

Jamsandekar, P. P., & Patil, M. K. (2013). Online learning - CBR approach. *International Journal of Research in Computer Science and Information Technology , 1* (A), 111-113.

Kolodner, J. (1993). Case-Based Reasoning. *Morgan Kaufmann Publishers* .

Kolodner, J. L., Cox, M. T., & Gonzales-Calero, P. A. (2005). Case-based reasoning-inspired approaches to education. *The Knowledge Engineering Review , 20*, 299-303.

Kolodner, J. L., Owensby, J. N., & Guzdial, M. (2004). Case-based learning aids. (D. H. Jonassen, Éd.) *Handbook of Research for Educational Communications and Technology* , 829-861.

Lieber, J. (2007). Application of the Revision Theory to Adaptation in Case-Based Reasonning: the Conservative Adaptation. *4626*.

Matveev, L. P. (1965). Periodization of sports training. *Fizkultura i Sport* .

Plaza, E., & Mc Ginty, L. (2005). Distributed case-based reasoning.

Rishi, O. P., Govil, R., & Sinha, M. (2007, October 24-26). Agent based student modeling in distributed CBR based intelligent tutoring system. *Proceeding of the World Congress on Engineering and Computer Science WCECS 2007* , 473-477.

Rishi, O. P., Govil, R., & Sinha, M. (2007). Distributed case-based reasoning for intelligent tutoring system: An agent based student modeling paradigm. *World Academy of Science, Engineering and Technology , 5*, 273-276.

Ronnestad, B. R., Ellefsen, S., Nygaard, H., Zacharoff, E. E., Vikmoen, O., Hansen, J., et al. (2012). Effects of 12 weeks of block periodization on performance and performance indices in well-trained cyclists. *Scandinavian Journal of Medicine and Science in Sports , 24* (2), 327-335.

Schank, R. C., Fano, A., Bell, B., & Jona, M. (1994). The design of goal-based scenarios. *Journal of the Learning Sciences , 3* (4), 305-346.

Tan, X. H., Shen, R. M., & Wang, Y. (2012). Personalized course generation and evolution based on genetic algorithms. *Journal of Zhejiang University-SCIENCE C (Computers and Electronics) , 13* (12), 909-917.